

# Évaluation pratique 3

## CRIBLE D'ÉRATOSTHÈNE

### DESCRIPTION DU PROGRAMME

Le programme à réaliser doit être capable d'afficher les **nombre premiers parmi les entiers allant de 0 à  $n$** . L'entier  $n$  doit être saisi par l'utilisateur. Pour rappel un nombre premier est un entier naturel qui admet uniquement deux diviseurs (1 et lui-même). Par exemple, si  $n = 18$ , alors les nombres premiers affichés sont : 2, 3, 5, 7, 11, 13, 17.

Pour ce faire, vous devez utiliser la **méthode du crible d'Ératosthène**. Cette méthode consiste à éliminer tous les multiples d'un entier naturel. Ainsi, ne subsistent que les entiers n'étant multiples d'aucun entier, autrement dit les nombres premiers.

Ce sont tout d'abord les multiples de 2 qui sont éliminés (à l'exception de lui-même). Puis, c'est ensuite au tour de chaque entier qui n'a pas été précédemment éliminé. L'**élimination des multiples d'un entier** débute par le carré de cet entier. Si ce carré est plus grand que  $n$ , alors le processus se termine.

Par exemple, pour les entiers naturels de 2 à 10 :

- a) élimination des multiples de 2 en débutant par 4

2	3	<del>4</del>	5	<del>6</del>	7	<del>8</del>	9	<del>10</del>
---	---	--------------	---	--------------	---	--------------	---	---------------

- b) élimination des multiples de 3 en débutant par 9

2	3	<del>4</del>	5	<del>6</del>	7	<del>8</del>	<del>9</del>	<del>10</del>
---	---	--------------	---	--------------	---	--------------	--------------	---------------

Dans ce cas, l'élimination des multiples de 5 n'a pas lieu car  $5^2$  est au-delà de 10. Les entiers premiers de 2 à 10 sont donc : 2, 3, 5, 7.

## EXEMPLE D'EXÉCUTION

Vous devez respecter la mise en forme proposée dans l'exemple ci-dessous. Les **données saisies par l'utilisateur** sont indiquées en bleu.

Nombres premiers de 0 à ... ? **50**

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

## STRUCTURE DE DONNÉES

Vous devez utiliser un tableau de  $n + 1$  booléens afin d'y mémoriser l'état des entiers naturels allant de 0 à  $n$ , plus précisément : **true** pour *conservé* et **false** pour *éliminé*.

Par exemple, si  $n$  vaut 10, alors le tableau qui résulte de l'application de la méthode du crible d'Ératosthène est le suivant :

<b>false</b>	<b>false</b>	<b>true</b>	<b>true</b>	<b>false</b>	<b>true</b>	<b>false</b>	<b>true</b>	<b>false</b>	<b>false</b>	<b>false</b>
0	1	2	3	4	5	6	7	8	9	10

La valeur **false** à la position 4 indique que l'entier 4 correspondant a été éliminé.

Initialement, le tableau doit contenir uniquement des valeurs **true**, sauf aux positions 0 et 1. En effet, les deux entiers correspondants sont d'office éliminés.

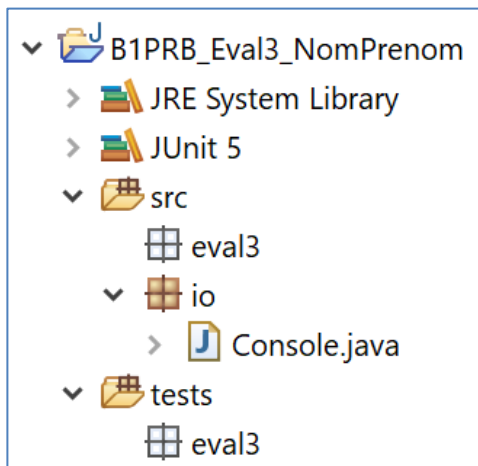
Par exemple, si  $n$  vaut 10 :

<b>false</b>	<b>false</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>
0	1	2	3	4	5	6	7	8	9	10

**[SUGGESTION]** Pour réaliser cette initialisation, aidez-vous de la fonction **Arrays.fill**.

## MISE EN PLACE DANS ECLIPSE

1. Dans Eclipse, créez un projet Java nommé **B1PRB\_Eval3\_NomPrenom**, où **NomPrenom** doit être remplacé par vos nom et prénom (par exemple, **DoeJohn**).
2. Reproduisez exactement l'organisation suivante :



## PHASE PRÉPARATOIRE

Pour cette évaluation, il est recommandé (mais non obligatoire) de réaliser une phase préparatoire pour la fonction décrite ci-dessous. Au moment du dépôt, vous pouvez l'ajouter à l'archive ZIP, mais cette dernière ne sera pas comptabilisée dans l'évaluation.

## CONSIGNES

1. Dans le package **eval3** du dossier des sources **src**, créez une classe nommée **CribleEratosthene** comportant une fonction **main**.
2. Dans le package **eval3** du dossier des sources **tests**, créez une classe pour les tests unitaires nommée **CribleEratostheneTest**.

### REMARQUE

Pour créer cette classe, faites un clic droit sur le package **eval3** du dossier des sources **tests**, puis sélectionnez **New > JUnit Test Case**.

3. Dans la classe **CribleEratosthene**, déclarez une fonction nommée **eliminerMultiples** qui élimine tous les entiers naturels multiples de l'entier spécifié

(à l'exception de lui-même) en définissant à **false** les valeurs booléennes correspondantes au sein d'un tableau :

```
public static void eliminerMultiples(boolean[] naturels, int a)
```

Le paramètre `naturels` est un tableau répertoriant les états (conservé ou éliminé) des entiers naturels allant de 0 à  $n$ .

Le paramètre `a` est l'entier pour lequel les multiples doivent être éliminés.

4. Dans la classe `CribleEratostheneTest`, implémentez des tests assurant au minimum 90% de **couverture du code** de la fonction `eliminerMultiples`.
5. Ajoutez un commentaire de type **javadoc** avant la déclaration de la fonction `eliminerMultiples` en donnant une brève description du traitement de cette fonction et en complétant correctement les tags.
6. Codez le reste de la solution dans la fonction `main` à l'aide de la fonction `eliminerMultiples`.

Veillez à respecter les bonnes pratiques de programmation (mise en forme du code, commentaires, noms adaptés pour les variables ...).

**[IMPORTANT #1]** Utilisez la classe `io.Console` pour effectuer les acquisitions.

**[IMPORTANT #2]** Réalisez une validation de chaque donnée saisie par l'utilisateur. Le cas échéant, l'utilisateur doit recommencer la saisie.

## DÉPÔT VIA HELMO LEARN

Créez une **archive ZIP** contenant le dossier principal de votre projet *Eclipse*. L'archive doit porter le même nom que votre projet.

**[IMPORTANT]** Le non-respect des consignes entraîne une pénalité de points.